N91-20678

AN INTELLIGENT VALUE-DRIVEN SCHEDULING SYSTEM FOR SPACE STATION FREEDOM WITH SPECIAL EMPHASIS ON THE ELECTRIC POWER SYSTEM

Presented by: Joseph C. Krupp

Decision-Science Applications, Inc.
1110 North Glebe Road, Suite 400
Arlington, Virginia 22201

ABSTRACT:

This paper discusses the Electric Power Control System (EPCS) created by Decision-Science Applications, Inc. (DSA) for Lewis Research Center (LeRC). This system in its current form makes decisions on what to schedule and when to schedule it, including making choices among various options or ways of performing a task. The system is goal directed and seeks to shape resource usage in an optimal manner using a value-driven approach. The paper discusses the considerations governing what makes a "good" schedule; how to design a value function to find the best schedule; and how to design the algorithm which finds the schedule that maximizes this value function. Results are shown which demonstrate the usefulness of the techniques employed. The value-driven approach also allows for the system to be easily extended to an emergency response system, making decisions as to where to best cut power when warranted.

1.0 DEFINITIONS

1.1 Activities

The EPCS schedules **activities, tasks, options**, and **subtasks**. These terms have very specific meanings with regard to the scheduler and are defined as follows:

Activity. A group of tasks directed toward a single goal, e.g., "Core Activities" or "Biology Experiments". Each activity has a value or priority.

Task. A well defined part of an activity, e.g., "Metallurgy Experiment 1". Each task is independent of all others, i.e., there is no specific order in which the various tasks must be completed. The tasks may even be done simultaneously. However, each task can have a time window, meaning that the task must be performed sometime within a particular time interval. The time window is not due to dependence of the tasks, but rather due to the nature of the task (e.g., it must be done during an eclipse period). [Actually, the time window is associated with the "option"--see below.] A task may be either a single time task, or may be a task which should be repeated periodically. The period of a task may be defined in terms of hours, orbits, or days. Each task has a value associated with it expressed as a percentage of the value of the activity of which it is a part. The highest priority task always receives 100% of the activity value. Less desirable tasks may receive a lower percentage of the activity value.

Option. A way of performing a task. Different options may have different numbers of subtasks, and will usually have different resource profiles. The options may also have different time windows. For example, a surveillance activity may have several windows of opportunity during which the surveillance can occur. Only one option for a given task is scheduled. Each option has a value associated with it expressed as a percentage of the value of the task. The best option always receives 100% of the task value. Less desirable options receive a lower percentage of the task value. For real time control (i.e., emergency response) it is important to know not just the value of a completed option, but how this value accrues. The actual value accrued as the option is performed depends on how much of the option is completed. Each option has a defined function describing the amount of value obtained (as a percentage of total value) as a function of percent completion of the option. The fraction of completion is based on the currently completed subtasks associated with the option.

Subtask. A well defined part of an option. The subtasks must be performed in a particular order. However, the time between subtasks may be variable. There may be a wait period before which the next subtask cannot be started, and a relative time window in which the next subtask must be completed. Each subtask is classed as non-restartable, restartable, or interruptible. If a non-restartable subtask is aborted, the task (option) of which it is a part cannot be completed and all subsequent value associated with that task is lost. If a restartable subtask is aborted, then it may be restarted from the beginning as if it had never been scheduled in the first place (assuming it can do so within its time window). If an interruptible subtask is aborted, it may be restarted at the exact point it was aborted without loss. The percentage completion for the activity of which it is a part is based on the percent completion for an interruptible subtask. For other classes, the subtask must be completed before the percent completion is increased.

Note that the values associated with activities, tasks, and options are best thought of as being set independently as if by a Vice President (activities), a department head (tasks), and a project manager (options).

1.2 Resources

The EPCS currently recognizes three types of resources:

Assignables. Assignable resources are those which are used in discrete units and which can be reused by another subtask after being released by the subtask currently using them. Examples include crew and workstations.

Consumables. Consumable resources are those which are used in arbitrary amounts and which are destroyed (or created) on use. Consumables may be produced by a subtask instead of consumed. For example, electrolysis consumes water and electricity, but produces oxygen and hydrogen.

Specifics. Specific resources are those which are a particular kind of generic resource. For example, the crew may contain specialists. One activity may need a metallurgist while another may need any crew member. The metallurgist is a specific type of crew, which is a generic assignable. Using the metallurgist reduces both the number of metallurgists available and reduces the number of crew available.

Other types of resources may be defined but are not currently incorporated in the EPCS. For example, one type of resource is a "state". Some subtasks may generate a vibration state which prohibit certain other activities from functioning. This resource type is currently being considered for addition into the EPCS.

Note that electric power is both an assignable and a consumable. From the concept of *power*, it is an assignable--only so much power can be drawn at any time. From the concept of *energy*, it is a consumable, since the batteries can hold only so much energy.

2.0 WHAT CONSTITUTES A GOOD SCHEDULE

The *primary* purpose of the EPCS is to schedule activities in such a way that the productivity of Space Station Freedom (SSF) is enhanced. In its simplest form, this translates to solving a knapsack problem. That is, if one schedule allows a certain set of tasks to be performed and a second schedule, by moving the subtasks around, makes room for one more task to be performed, then the second is better. But this is a simplistic view of things, and in reality tradeoffs must be made. The first and most obvious tradeoff is that not all activities are as important as others. Thus, the notion of values comes into play. If values are assigned to the activities, tasks, and options, then that schedule which allows a set of activities with a total higher value than that of another set of activities is clearly better. So far, so good. But there are other tradeoffs.

All options have a time window associated with them (which may in fact be the entire planning time) during which they may be performed. Usually there is some preference as to when in this time window it would be better to schedule the option. In most cases, this is at the beginning of the window--due to the possibility of unforeseen problems, it is better to be early than late. Two schedules may schedule the exact same set of options. The first, however, may have all options being performed early in their time window while the second may have some options being performed late in their time window. One would judge the first schedule as better than the second. Thus, some value must be lost the longer an option is delayed.

In a similar vein, many tasks are periodic, i.e., they need to be scheduled on a regular basis. If a task is to be scheduled on a daily basis, one would prefer a schedule in which the task is performed at roughly the same time every day to one in which the task is performed late in the day one time and early in the day the next.

The scheduler must consider two aspects of the power system: battery charge and power flow. Both of these aspects are important. Consider two schedules which are identical with regard to the tradeoffs discussed above. If in the first the battery is drawn to a dangerous level of discharge while in the second the battery is always well charged, then the second is clearly better. Similarly, if the first has periods of very high power consumption followed by periods of very low power consumption, while the second maintains a relatively constant power flow, then the second is better. This is because I^2R losses for the first schedule will be higher.

Assigning values to the activities is not a problem affecting the design of the scheduler. The user may assign values to the various activities, tasks, and options in any way he wishes. The other tradeoffs do pose a problem, however, because a value function must be devised such that various tradeoffs are properly balanced. For example, if the battery charge and power flow considerations dominate, the best schedule may be to do nothing. Then the battery could stay happily charged and the power distribution system could stay cool. But this hardly enhances productivity! Similarly, the purpose of specifying a time window for an activity is that it is acceptable to delay the start of the activity, so the value lost by delay should not be great enough to prevent moving the activity in order to allow an additional activity to be scheduled.

3.0 COSTS VERSUS VALUES

The EPCS is a value-driven scheduler and emergency response system. By value-driven is meant that decisions are made which maximize total value returned, i.e., profit. In the previous sections we discussed the *intrinsic* values of an activity, task, and option, and discussed various tradeoffs that need to be considered in choosing one schedule over another. In this section we will describe the notion of *costs*, and the role they play in quantifying or codifying these tradeoffs. The value function to be maximized is represented as value minus costs, and it is the functional form of the costs which codify the tradeoffs to be made. Costs are of two types: 1) resource costs (the marginal cost of an additional unit of resource), and 2) opportunity costs (which relate to the time placement of the subtasks).

3.1 Resource Costs

3.1.1 Nominal Costs

All subtasks use resources. We would like to define the concept of cost for a resource in order to quantify the profit gained by performing a particular option. The cost concept is intuitive and easy to understand--options using more costly resources may be less preferred to options using less costly resources depending on the relative intrinsic value of the two options. But how do we define the cost of a resource? To a first approximation, each resource can be thought of as having a *nominal cost* inversely proportional to the amount normally used during a planning period. That is, if the Space Station is sized to use X units of nitrogen and Y man-days then the nominal cost of nitrogen is $1/X$ and the nominal cost of a man-day is $1/Y$. In this way, the total nominal cost of every

resource during a normal planning period is 1.0. This normalization reflects the fact that the Space Station was sized intelligently and provides an intuitive quantification of the marginal cost of a unit of resource *in the Space Station environment.*

It should be noted that in the real Space Station environment these nominal costs would be provided to the EPCS by the individual control systems. This will allow the individual systems to make allowances for special situations. For example, a consumable resource which was being used at a much slower rate than usual could have its nominal cost lowered, while one which was being used more quickly could have its nominal cost raised. In the prototype system, the nominal costs are set by considering the total amount of resource needed for all options as a surrogate for the amount normally used in a planning cycle.

3.1.2 Cost/Benefit Ratios

The use of nominal costs also allows for an interpretation of the intrinsic values of an option as a benefit/cost, or profit/cost, ratio. This ratio is more what the user has in mind when he assigns values to the various activities, tasks, and options. If one option has an intrinsic value twice that of another, the user expects that option to be scheduled over the other if possible. If the nominal costs (i.e., the sum of all resources used times their nominal costs) of the two options differ substantially, this may not be the case. The definition of the user-supplied values as profit/cost ratios lets the user assign these values without needing to know the nominal cost of performing the option. Thus, the actual value of an option used in the EPCS is the product of (one plus) the assigned value times the nominal cost of the option.

3.1.3 Cost Curves

Any resource which has a supply much larger than required to perform all subtasks is a non-player with regards to any scheduling decision which is made. In the real world, such a resource would be free ("You can't even give that stuff away."). If all resources were like this, one would maximize total value by scheduling the most preferred options of every task. This is not usually the case, however. Any resource which is in short supply will have a cost associated with it which reflects the balance of supply and demand. That is, there is a cost curve associated with each resource. Resources which use more than the nominal amount for the planning period will have a higher cost than resources which use less than the nominal amount for the planning period. The cost curves are supplied by the resource control systems aboard the Space Station. We will provide the cost curve for the electric power system in detail. In the prototype, an exponential is used as a surrogate cost function for the other resources:

$$\lambda = \lambda_0 \exp[\alpha (U-R)/R] \qquad (1)$$

where α = an adjustable parameter for each resource,
λ_0 = the nominal cost of the resource
U = the resource usage
and R = the amount of resource available (or nominal amount to be used during the planning cycle)

Note that for assignables, resource usage is in terms of units used at any given time, e.g., crew used minus total crew, while for consumables the resource usage is in terms of the largest deficit at any time in the future. That is, if a consumable is overused by tomorrow, I should conserve it today.

An option using a given resource will lose value equal to the cost of that resource. In fact, if the cost is so high that the total profit, value minus cost, is negative, it is preferable to drop the option from the schedule. Resource costs are therefore a function of resource use. At low use the costs are low, while at high use the costs are high. There are two aspects of these resource costs which need to be determined: nominal costs and cost curves.

Note that the cost curves do not usually go to infinity if the resource is overutilized. This is because the scheduling algorithm is an iterative one. Making the cost curves go to infinity at the constraint point, i.e., using a "brick wall" approach, will ensure feasibility but will not promote optimality. It is necessary to allow infeasibility so that a particular subtask which can move out of the way will do so. This will be made clear in a later section when we discuss the algorithm employed.

3.2 The Electric Power Cost Functions

3.2.1 The Battery Charge Penalty

We desire a battery charge penalty, or cost, designed to keep the battery reasonably well charged, yet still allow the battery to go below nominal minimum charge if absolutely necessary. This function should be zero at the charge level where trickle charging needs to begin, be 1.0 (times the nominal cost) at the nominal minimum discharge level, and rise sharply below this level. Such a function is simply:

$$\lambda_B = \lambda_0^{elec} [(C_{TR} - C) / (C - C_{min})]^\beta \qquad (2)$$

where C_{TR} = trickle rate level (.95),

λ_0^{elec} = the nominal cost of electricity (per kw-H)

C_{min} = $2 C_{nom} - C_{TR}$ (.35 for given C_{nom} and C_{TR})
= absolute minimum that will not be violated under any circumstances.

and C_{nom} = nominal minimum discharge level (.65).

This function is plotted in Figure 1 for $\beta = 3$.

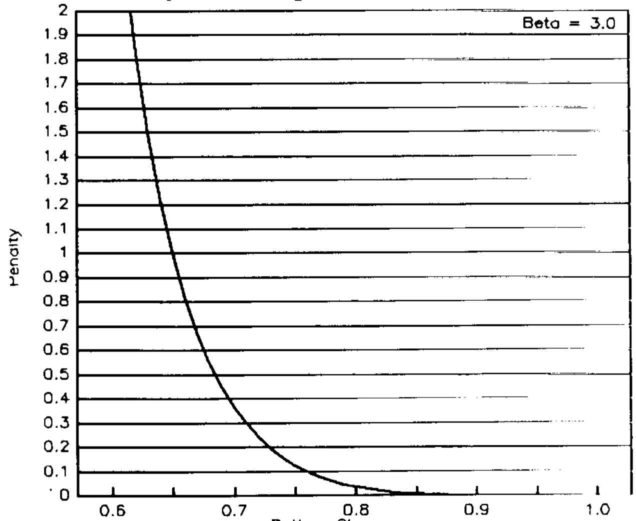

Figure 1: Battery Penalty Functions

Note that the total penalty is integrated over the entire planning time. Thus, a schedule which went below the nominal minimum discharge for a very brief period, but otherwise stayed well above it, would score better than one that stayed above the nominal minimum discharge level, but just barely above it, for a long period of time. This makes sense from an emergency response point of view.

3.2.2 The Power Flow Penalty

Power enters the scheduling considerations in two ways. For high power levels, the primary consideration is that the power not exceed acceptable safe levels. In fact, the power flow penalty should approach infinity if the power required is greater than the maximum power available (even if the battery has energy available, it is limited in how quickly it can deliver this energy, i.e., the power it can deliver is finite). For lower power loads, the primary consideration is to smooth out the power load across time. In general, this second consideration should be small enough that it does not cause a task to be removed from a schedule, but does cause it to be adjusted slightly to smooth the load.

The second consideration can be achieved through a modification of the electric power cost, λ_B, described above. Since the reason for wanting to balance the loads is to reduce I^2R losses, λ_B can be multiplied by a factor proportional to the square of the power as follows:

$$\lambda_B' = \lambda_B [1 + (\beta P)^2] \quad (3)$$

where $\beta = [(1 - \epsilon) / \epsilon]^{0.5} / P_0$,

ϵ = Efficiency of the PMAD system at the nominal power level (0.93)

and P_0 = the nominal power level

The first consideration is no different from any other type of resource. Here power is considered as an assignable resource--only so much power can be used at any given instant. We therefore use an equation similar to Eq. (1) above:

$$\lambda_P = \lambda_0^{elec} \exp[\alpha (P - .9 P_{PV}) / P_{PV}] \quad (4)$$

where P_{PV} = Power from the Photovoltaic Array.

3.3 Opportunity Costs

The schedule may be shaped by other considerations than how the resources are used. For example, there may be options for which there is a preference as to where in the time window the option is scheduled. Similarly, for tasks which need to be repeated on a periodic basis it is preferred that each subtask be performed more or less at the same time within each period. Although the prototype EPCS does not yet take into account any preference within the time window, it does take into account the preference for periodic tasks being performed at similar times within the period. It does so by subtracting from the profit of an option a fraction of each subtask's value[1], where the fraction is defined as:

$$1.0 - \exp[-0.5 ((t-t_0)/\sigma)^2], \quad (5)$$

where t = time of proposed scheduling of the subtask

t_0 = desired time of scheduling $= [(t_{-1} + T_R) + (t_{+1} - T_R)] / 2$

$\sigma = 2 T_R$

and T_R = Repeat time

i.e., a Gaussian centered around the desired time with a standard deviation of twice the repeat time.

This function is small enough that subtasks are free to move if necessary, yet large enough that, all things being equal, the subtasks will tend to be performed at regular times.

4.0 THE SCHEDULING ALGORITHM

The scheduling algorithm consists of three interacting processes:

1. The Basic Algorithm
2. Feasibility Adjustment
3. Optimality

4.1 The Basic Algorithm

The basic algorithm consists of scheduling each task in turn. During this phase, one option for each of the tasks is always scheduled. When scheduling a particular task, the scheduled option from the previous iteration is "picked up", i.e., is removed from the schedule. The resource usage is calculated with all other tasks implemented. This determines the resource cost for any subtask of any option for the task to be scheduled. Each option for the task is considered in turn and an optimal placement for all subtasks for that option is determined. The option with the largest profit (even if the profit is negative at this point) is scheduled.

Determining the optimal placement of the subtasks for an option is done with a dynamic programming algorithm. The cost for each subtask is determined for each of N delay times. By working backwards from the last subtask to the first, the optimal delays (within the time resolution of the N delays) can be determined for the entire option.

4.2 Feasibility Adjustment

When the basic algorithm has converged, i.e., the options picked and the scheduled times for all subtasks is not changing from iteration to iteration, the schedule may not be feasible. There are two reasons for this. The primary reason is the granularity of time periods. Resource usage is not stored on a minute-by-minute basis for the entire planning time. Rather, a set of time periods, or "bins", are defined and resource usage is added to these bins. At the

[1]The value of a subtask is the value of the option times the ratio of the duration of the subtask to the sum of the duration for all of the subtasks. That is, for the purpose of the scheduling function of the EPCS we are assuming linear value accrual.

start of the program these time periods are defined as the daylight and eclipse times of the Space Station. Usage of assignables is by units times time, i.e., man-days, kW-days, workstation-days, etc. Therefore, while a time period may contain enough crew-days to satisfy the resource usage required for the schedule, there may be a conflict in that for a short period of time, more crew than are available are needed. To fix this problem, a rule based "tweak" algorithm is applied to make small adjustments to the schedule to get feasibility if possible, and new time periods are introduced at the problem spots to keep the problem from reoccurring.

4.3 Optimality

A secondary reason for non-feasibility, of course, is that more is being scheduled than can fit within available resource and time constraints. Thus, if the feasibility adjustment does not result in a feasible schedule, the system checks for negative profit. All tasks with a negative profit are sorted from most negative to least negative. The worst task is then removed from the schedule and the resource usage adjusted. If the next task now has a positive profit, it is skipped; otherwise it too is removed. This process continues until all tasks have a positive profit. Those tasks removed will stay removed (unless a second option could be scheduled). If the schedule is still not feasible, a call is made to the resource suppliers to adjust the cost curves[2] and the model resumes with the basic algorithm.

If the schedule is feasible, an endgame phase is entered whereby the schedule is adjusted according to the basic algorithm, but allowed to move only a few minutes either way from the current schedule. This is to make fine adjustments due to the opportunity costs associated with the timing.

5.0 RESULTS

Figure 2 shows the electric power profile for a particular two day schedule. The gray areas are eclipse periods. This schedule consisted of 10 activities containing 37 tasks which had 57 options consisting of 370 subtasks. The important thing to notice is that the power used, in a gross sense, is fairly uniform, and where there are peaks, they tend to fall during periods of daylight. Figure 3 shows the battery charge for this same schedule. In Figures 4 and 5, we have plotted graphs for the same mission which were derived by removing the cost of electric power. Note that we still did not let the battery become too discharged, but the power levels are grossly non-optimal. This demonstrates the technique employed, while providing feasible schedules at a minimum, also provides an efficient means for shaping resource usage.

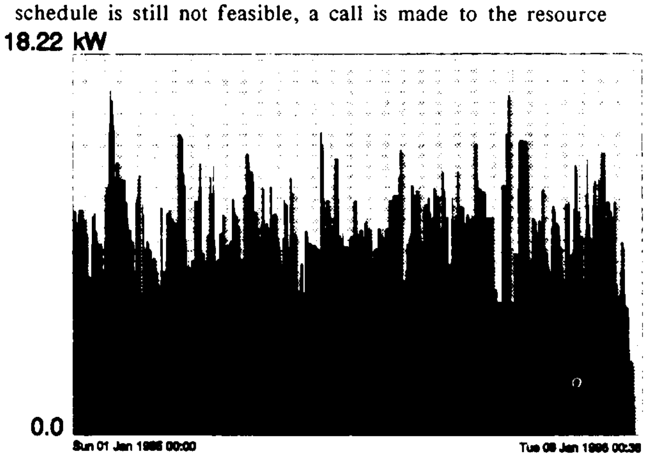

Figure 2. Electric Power Profile of Generated Schedule

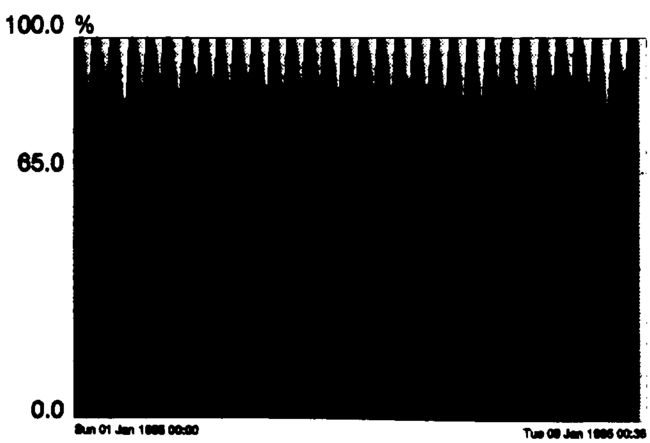

Figure 3. Battery Charge Profile of Generated Schedule

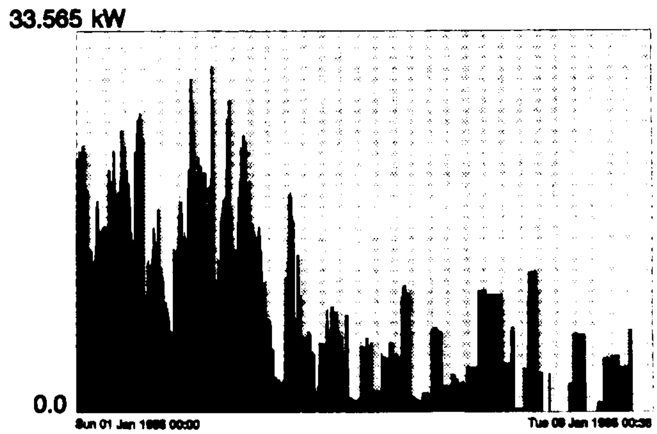

Figure 4. Electric Power Profile-No Resource Shaping

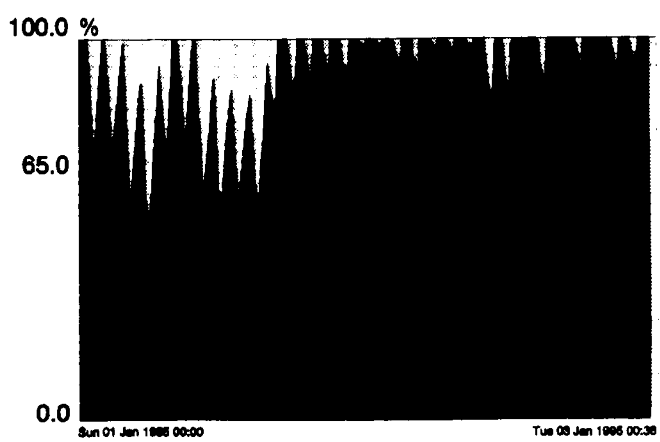

Figure 5. Battery Charge Profile-No Resource Shaping

[2]In the prototype, the value of α in Equation 1 is increased.